**String methods**

April 7, 2017 · 1,327 views

ในบทก่อนหน้า เราได้พูดเกี่ยวกับการใช้งาน String ในเบื้องต้นไปแล้ว ในบทนี้ เราจะแนะนำเมธอดและการใช้งานเมธอดในคลาส String ในภาษา Python นั้น String เป็นคลาสมาตรฐานสำหรับจัดการข้อมูลประเภทข้อความ และในคลาสจะมีเมธอดสำหรับจัดการข้อมูลและอำนวยความสะดวกในการเขียนโปรแกรมเป็นอย่างมากและช่วยลดเวลาในการเขียนโปรแกรมของคุณ

**Find and search**

ในคลาสของ String มีเมธอดประเภทต่างๆ ในการจัดการกับ String เมธอดกลุ่มแรกที่เราจะแนะนำให้คุณรู้จักคือเมธอดเกี่ยวกับการค้นหาและตรวจสอบตำแหน่งภายใน String มาดูตัวอย่างของการใช้เมธอดดังกล่าวในภาษา Python

s = 'mountain'

print(s.startswith('moun'))
print(s.endswith('in'))
print(s.find('n'))
print(s.rfind('n'))
print(s.replace('moun', 'con'))
print(s.count('a'))

ในตัวอย่าง เป็นการใช้งานเมธอดในการค้นหาและการทำงานกับตำแหน่งใน String โดยเมธอด startswith() เป็นเมธอดในการตรวจสอบว่า String นั้นเริ่มต้นด้วย 'moun' หรือไม่ และเมธอด endswith() เพื่อตรวจสอบว่า String นั้นจบด้วย 'in' หรือไม่ เพราะว่าข้อมูลใน String นั้นตรงกับเงื่อนไขดังกล่าวซึ่งมีค่าเป็นจริง

print(s.find('n'))
print(s.rfind('n'))

ต่อไปเป็นการใช้เมธอด find() เพื่อค้นหาตำแหน่ง Index ของ Sub string 'n' ที่พบครั้งแรกใน String และเมธอด rfind() เป็นการค้นหาจากด้านหลังจากตำแห่งแรกที่พบเช่นกัน ซึ่งจะได้ Index เป็น 3 และ 7 ตามลำดับ

print(s.replace('moun', 'con'))
print(s.count('a'))

และเมธอด replace() เป็นการใช้สำหรับแทนที่คำใน Srting โดยเราได้ทำการแทนที่ 'moun' ด้วย 'con' ซึ่งถ้าหากคำดังกล่าวพบมากกว่าหนึ่งครั้งใน String จะทำการแทนที่ทั้งหมดที่เจอ และเมธอด count() ใช้สำหรับนับตัวอักษรที่ระบุที่พบใน String เราได้ทำการนับว่าพบตัวอักษร 'a' เป็นจำนวนเท่าไหร่ใน String

True
True
3
7
contain
1

นี่เป็นผลลัพธ์การทำงานของโปรแกรม ในการใช้เมธอดเกี่ยวกับการค้นหาและทำงานในตำแหน่งของ String

**Text and sentence**

ในภาษา Python คลาสของ String ยังมีเมธอดในการทำงานเกี่ยวกับ text และข้อความ เมธอดเหล่านี้มักจะใช้สำหรับการจัดแสดงผลข้อความในโปรแกรมหรือเว็บไซต์ให้อยู่ในรูปแบบเดี่ยวกัน เช่น การเปลี่ยนตัวแรกของประโยคให้เป็นตัวพิมพ์ใหญ่ หรือแสดงข้อความในตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ เป็นต้น มาดูตวอย่างการใช้งาน

s = 'this is a game that i love to play.'

print(s.capitalize())
print(s.title())
print(s.upper())
print('MARCUSCODE.COM'.lower())

ในตัวอย่าง เป็นการใช้งานเมธอดต่างๆ ในการจัดการกับข้อความ เมธอด capitalize() ใช้สำหรับทำให้ตัวอักษรตัวแรกของ String เป็นตัวพิมพ์ใหญ่ ในขณะที่เมธอด title() นั้นจะทำให้ตัวแรกในทุกๆ คำที่คั่นด้วย white space เป็นตัวพิมพ์ใหญทั้งหมด

สำหรับเมธอด upper() แปลงให้ตัวอักษรทุกตัวใน String เป็นตัวพิมพ์ใหญ่ และเมธอด lower() แปลงให้ตัวอักษรทุกตัวใน String เป็นตัวพิมพ์เล็ก

This is a game that i love to play.
This Is A Game That I Love To Play.
THIS IS A GAME THAT I LOVE TO PLAY.
marcuscode.com

นี่เป็นผลลัพธ์การทำงานของโปรแกรม ในการใช้งานเมธอดจัดการกับข้อความเกี่ยวกับตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก

**Formatting string**

ในภาษา Python นั้นคลาสของ String ยังมีเมธอดในการจัดรูปแบบของ String สำหรับการแสดงผลอีกด้วย เช่น การเติมตัวอักษรเข้าไปใน String หรือการลบตัวอักษรที่ไม่ต้องการออกไปจาก String มาดูตัวอย่างการใช้งาน

s = 'Python'
print(s.center(10))
print(s.ljust(10))
print(s.rjust(10))
print(s.center(10, '#'))
print(s.ljust(10, '#'))
print(s.rjust(10, '#'))

number = '83'
print(number.zfill(6))
print('-1.32'.zfill(8))

ใน 3 เมธอดแรก คือเมธอดที่ใช้ในการเติมตัวอักษรเข้าไปในข้อความให้ได้ตามจำนวนที่ต้องการ เมธอด center() ใช้สำหรับเติมข้อความเข้าไปทางซ้ายและทางขวาให้ String มีความยาวกับที่ระบุ ส่วนเมธอด ljust() นั้นทำงานเช่นเดียวกันแต่เติมตัวอักษรเฉพาะทางซ้าย และเมธอด rjust() เติมตัวอักษรเพียงด้านขวา อาร์กิวเมนต์แรกเป็นจำนวนของ String ทั้งหมด ส่วนอาร์กิวเมนต์ที่สองเป็นทางเลือกโดยเป็นตัวอักษรที่ต้องการเติมลงไป และมีค่า Default เป็น Space และเมธอด zfill() เป็นการใช้เติมตัวเลข 0 โดยเรามักจะใช้กับ String ที่มีรูปแบบเป็นตัวเลข

 Python
Python
 Python
##Python##
Python####
####Python
000083
-0001.32

นี่เป็นผลลัพธ์การทำงานของโปรแกรมในการใช้เมธอดสำหรับเติมข้อความ ในกลุ่มคำสั่งแรกเป็นการเติมช่องว่างลงไปทางซ้ายและขวาของ 'Python' และหลังจากนั้นเติมเครื่องหมาย '#' และสุดท้ยเติมเลข 0 ลงไปข้างหน้าของ String ในรูปแบบตัวเลข

s = ' Python '
print(s.strip())
print(s.lstrip())
print(s.rstrip())

s2 = '----marcuscode----'
print(s2.strip('-'))
print(s2.lstrip('-'))
print(s2.rstrip('-'))

นอกจากเมธอดในการเติมข้อความล้ว ยังมีเมธอดในการตัดคำออกจากข้อความด้วยเช่นกัน โดยเมธอด strip() ใช้สำหรับตัดลำดับของตัวอักษรที่พบในทางซ้ายและทางขวาออกไป เมธอด rstrip() นั้นจะตัดเพียงทางขวา และเมธอด lstrip() จะตัดเพียงทางซ้าย อาร์กิวเมนต์เป็นตัวอักษรที่คุณต้องการตัด และมีค่า Default เป็น Space ถ้าหากคุณเว้นว่างไว้

Python
Python
 Python
marcuscode
marcuscode----
----marcuscode

นี่เป็นผลลัพธ์การทำงานของโปรแกรม ในกลุ่มคำสั่งแรกเป็นการตัดช่องว่างออกไปจากทางซ้ายและขวาของ String และกลุ่มต่อไปเป็นการตัดตัวอักษร - ออกไปจากทางซ้ายและขวาของ String

ในภาษา Python ยังมีเมธอด format() สำหรับจัดรูปแบบการแสดงผลของ String ที่เรียกว่า String interpolation การทำงานของเมธอดนั้นจะคล้ายกับการจัดการแสดงผลในฟังก์ชัน print() มาดูตัวอย่างการใช้งาน

print('a = {0}'.format(3))
print('{0} loves to learn {1}'.format('Mateo', 'Python'))

ในการใช้งานเมธอด format() นั้นเราต้องกำหนดรูปแบบการแสดงผลให้กับ String ในรูปแบบ {n} โดยที่ n นั้นเป็นลำดับของอาร์กิวเมนต์ที่ส่งเข้าไปในเมธอดซึ่งเป็นจำนวนเต็มที่เริ่มจาก 0 ไปเรื่อยๆ ในคำสั่งแรกนั้นมีเพียงหนึ่งอาร์กิวเมนต์ และในคำสั่งที่สองนั้นมีสองอาร์กิวเมนต์ เราจะเพิ่ม {1} เข้ามาสำหรับอาร์กิวเมนต์ที่สอง

a = 3
Mateo loves to learn Python

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

**Type checking**

เมธอดที่สำคัญในคลาส String อีกกลุ่มหนึ่งคือเมธอดในการตรวจสอบประเภทข้อมูลภายใน String ว่าเป็นประเภทที่กำหนดหรือไม่ เช่น การตรวจสอบหาว่ามีตัวเลขภายใน String หรือไม่ หรือตรวจสอบว่า String นั้นเป็นตัวเลขหรือไม่ เป็นต้น มาดูตัวอย่างการใช้งาน

# tesing true
print('marcuscode2017'.isalnum())
print('bobby'.isalpha())
print('1234'.isdecimal())
print('1234'.isdigit())
print('1234'.isnumeric())
print('mateo'.islower())
print('MATEO'.isupper())

# tesing false
print('marcuscode 2017'.isalnum()) # has space
print('bobby1'.isalpha()) # has number
print('7F'.isdecimal()) # F is not decimal
print('1234A'.isdigit()) # A is not digit
print('Python'.isnumeric()) # not number
print('Mateo'.islower()) # M is upper case
print('Mateo'.isupper()) # Not all are upper case

ในตัวอย่าง เป็นทดสอบการตรวจสอบ String ด้วยเมธอดต่างๆ สำหรับการตรวจสอบที่เป็นจริงและไม่เป็นจริง สำหรับการทดสอบที่ไม่เป็นจริงนั้นเราได้คอมเมนต์เหตุผลที่ทำให้ไม่เป็นจริงด้วย และนี่เป็นผลลัพธ์การทำงานของโปรแกรม

True
True
True
True
True
True
True
False
False
False
False
False
False
False

**List and sequence methods**

นอกจากนี้ในคลาส String ยังมีเมธอดที่ทำงานเกี่ยวกับ List ในการรวม String จากลิสต์เข้าด้วยกันด้วยเมธอด join() หรือการตัดคำออกจาก String ด้วยเมธอด split() หรือเมธอด splitlines() นี่เป็นตัวอย่างการใช้งานเมธอดเหล่านี้

countries = ['England', 'Turkey', 'Poland', 'Russia', 'Germany']
print(', '.join(countries))

lang = 'Java PHP C++ Python C#'
print(lang.split(' '))

text = 'Python is a language\nused to create a web,\n'
text += 'desktop application\nand more'
text2 = text.splitlines()
for t in text2:
 print(t)

ในตัวอย่าง เรามีตัวแปรลิสต์ countries สำหรับเก็บข้อมูลของประเทศต่างๆ เราใช้เมธอด join() เพื่อรวม String ในลิสต์เข้าด้วยกันโดยคั่นด้วย ', ' และในตัวแปร lang นั้นเป็นข้อความที่คั่นด้วยช่องว่าง เราใช้เมธอด split() เพื่อแยก String ออกจากตัวแปรและจะได้ผลลัพธ์เป็นลิสต์ คุณสามารถใช้ตัวอักษรอื่นได้โดยใส่ในอาร์กิวเมนต์ของเมธอด สำหรับเมธอด splitlines() เป็นการแบ่ง String ด้วยการขึ้นบรรทัดใหม่

England, Turkey, Poland, Russia, Germany
['Java', 'PHP', 'C++', 'Python', 'C#']
Python is a language
used to create a web,
desktop application
and more

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

ในบทนี้ คุณได้รู้จักกับการใช้งานเมธอดจากคลาส String ในรูปแบบต่างๆ เช่น เมธอดในการการค้นหาข้อความและ Index ของ String เมธอดในการทำงานเพื่อแปลง String เป็นตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ เมธอดที่ใช้ในการจัดรูปแบบของ String เมธอดในการตรวจสอบประเภทและข้อมูลใน String เป็นต้น ซึงเราได้แนะนำให้คุณรู้จักในเบื้องต้น สำหรับเมธอดทั้งหมดคุณสามารถดูได้ที่เอกสารของเว็บไซต์ภาษา Python ที่ <https://docs.python.org/3/library/stdtypes.html#string-methods>